Huffman encoding and priority queue
Due: Monday April 22 11:59PM
Total: 20 points

Unless explicitly stated, all the assignments can be worked by a team of 2 students.


In this assignment, you will write HuffmanTree.java to construct Huffman tree for given symbols and frequencies. The Huffman Tree contains nodes that are instances of class HuffmanNode. The class skeletons, driver, sample inputs and outputs are provided. You will use PriorityQueue when building the tree. You can use java.util.PriorityQueue<E> class in Java or implement it yourself. In case that you use java.util.PriorityQueue, use methods add() for adding to the priority queue and poll() for removing from the priority queue.

Implementation:
Specially, you will define class HuffmanNode and implement its methods, and implement three methods in class HuffmanTree.

1. [4pts] define class HuffmanNode, instance variables, constructor, accessor and modifier methods.
2. Implement three methods of class HuffmanTree:
   a. [4pts] initQueue(String Filename). Read the characters and frequencies from the file and generate a priority queue containing n trees where n is the number of characters in the input file. Each tree has a single node with the character as the label and frequency as the weight.
   b. [4pts] buildHuffmanTree(). Use n-1 steps to build the Huffman tree from the initial priority queue built in initQueue().
   c. [4pts] printCodewords(). Use prefix traversal to obtain the codeword for each character and print it.

[4pts] Input and output:
You program should work with the provided driver and take a filename as a command argument. For example,

Java HuffmanTree huff.txt

There is one character and its weight delimited by “:” in each line of huff.txt. The content of huff.txt looks like:

<table>
<thead>
<tr>
<th>Character</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
</tr>
</tbody>
</table>

Your program prints out the codewords to console. The corresponding output for huff.txt is:
The Huffman codewords for the alphabet and weights specified in huff.txt is:
A: 0
C: 1000
D: 1001
E: 1010
F: 1011
G: 1100
H: 1101
B: 111

You are supposed to generate more input files for testing. Turn in HuffmanTree.java through turnin system before due time.